 Ingeniería Informática del Software

**Diseño y Pruebas**

**Informe jMeter – D10: Item 5.**

Barrientos Mohedano, Rubén

Egea Guerrero, Simón

García da Silva, Felipe Javier

Lorenz Rosado, Nicolás

Índice

[**Introducción** **2**](#_Toc7804)

[**Análisis máximo rendimiento** **3**](#_Toc7805)

[**Register** **3**](#_Toc7806)

[**Edit personal data** **4**](#_Toc7807)

[**Browse chorbies** **5**](#_Toc7808)

[**Browse chorbies who like them** **6**](#_Toc7809)

[**Like and cancel like** **7**](#_Toc7810)

[**Edit searchTemplate and browse results** **9**](#_Toc7811)

[**Write chirp** **10**](#_Toc7812)

[**Forward chirp** **11**](#_Toc7813)

[**Reply chirp** **12**](#_Toc7814)

[**Delete chirp** **13**](#_Toc7815)

[**Ban and unban chorbi** **14**](#_Toc7816)

[**Edit cache time** **15**](#_Toc7817)

[**Change banners** **16**](#_Toc7818)

[**Dashboards** **17**](#_Toc7819)

[**Conclusiones** 18](#_Toc7820)

# Introducción

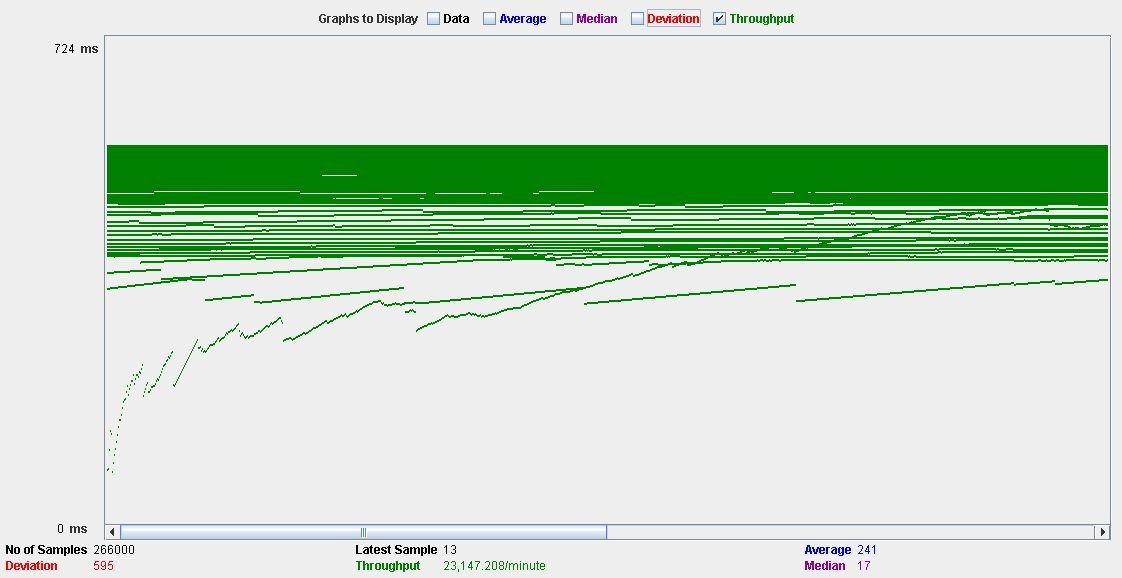
En las próximas páginas se mostrará un análisis de las pruebas realizadas con la herramienta jMeter sobre nuestro proyecto.

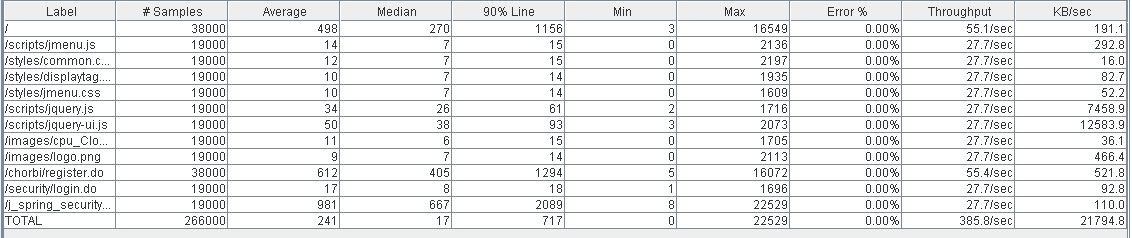
Se destacará cual es el máximo rendimiento del sistema y se mostrarán capturas que lo confirmen.

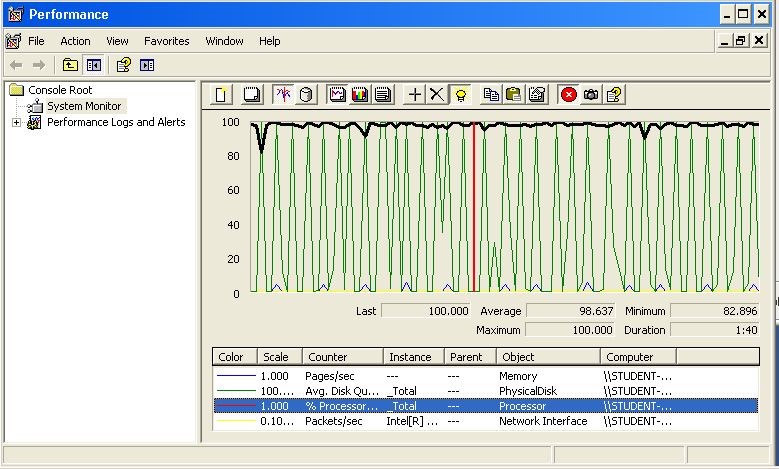
# Análisis máximo rendimiento

Mediante el uso de JMeter hemos obtenido un script para cada caso de uso. Para cada uno de ellos se han realizado pruebas primero comenzando con 10,20,50,100,150 y 200 usuarios. Al usar una carga de 200 usuarios ya comenzaba a dar fallos en la mayoría de casos de uso, pero al reducirlo a 190 aun funcionaban todos de forma correcta. A continuación, vamos a ver las capturas de cada caso de uso con una carga máxima de 190 usuarios.

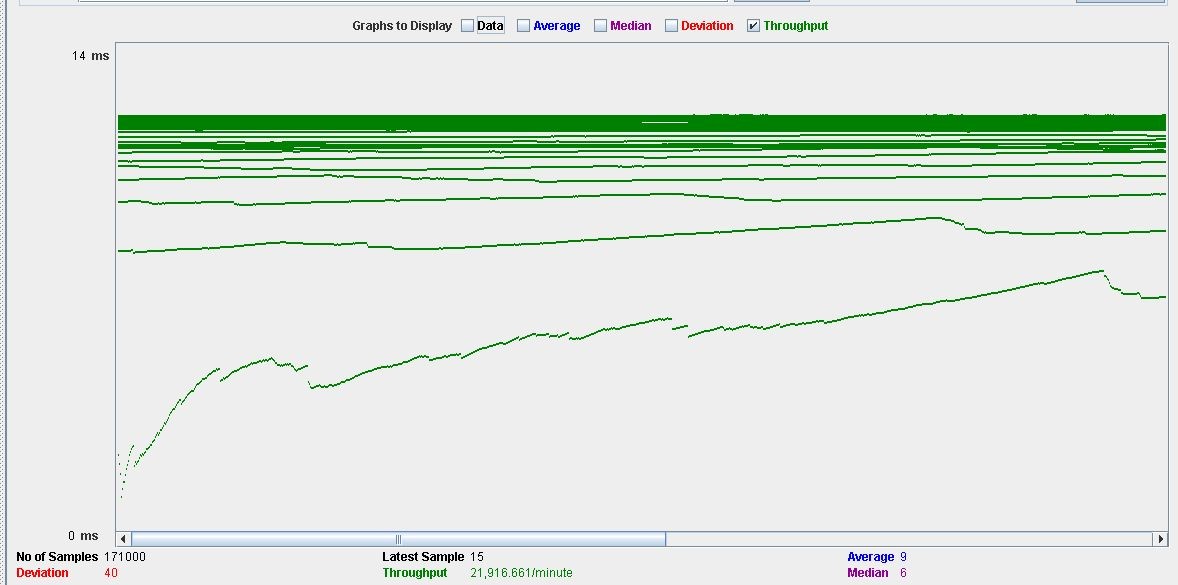
# Register

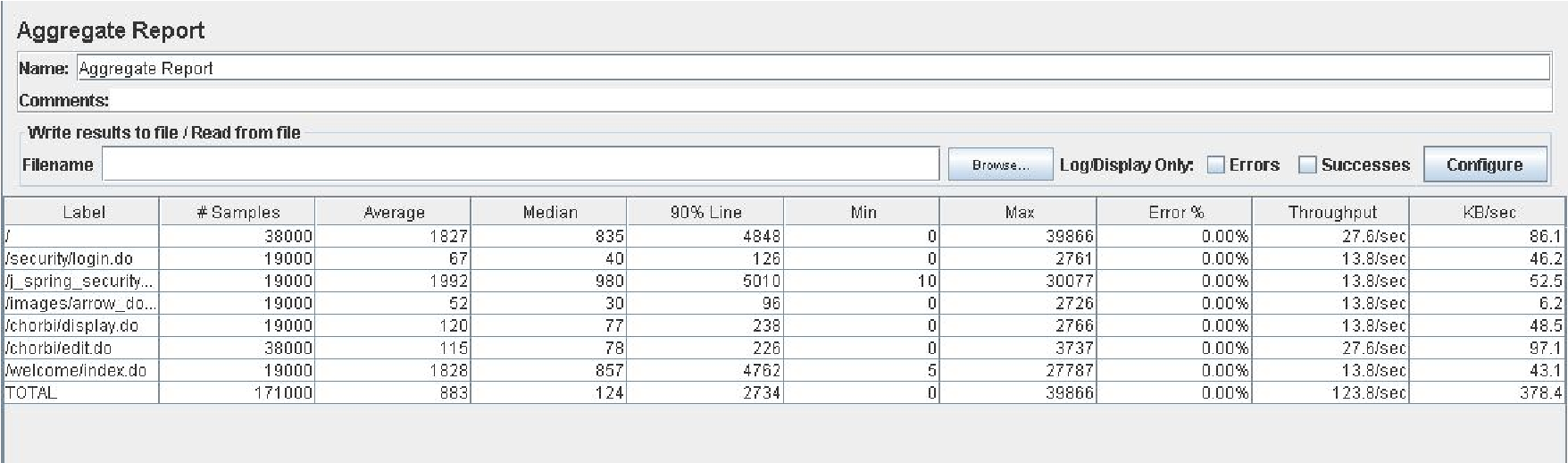


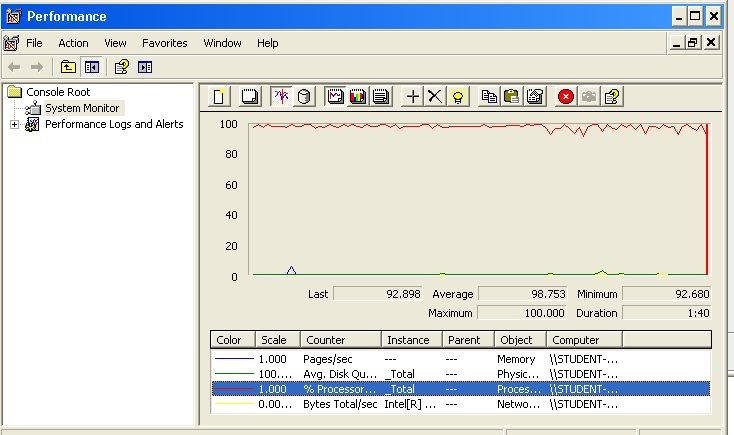




# Edit personal data

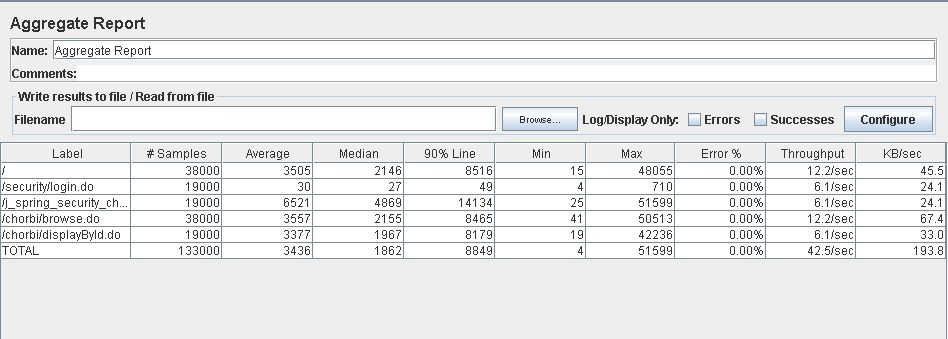


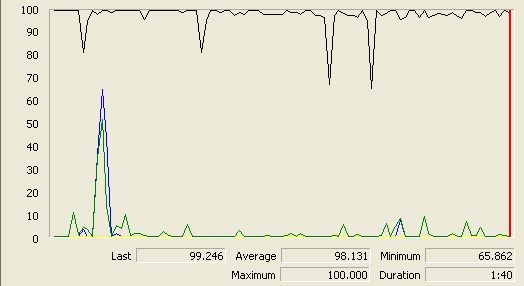




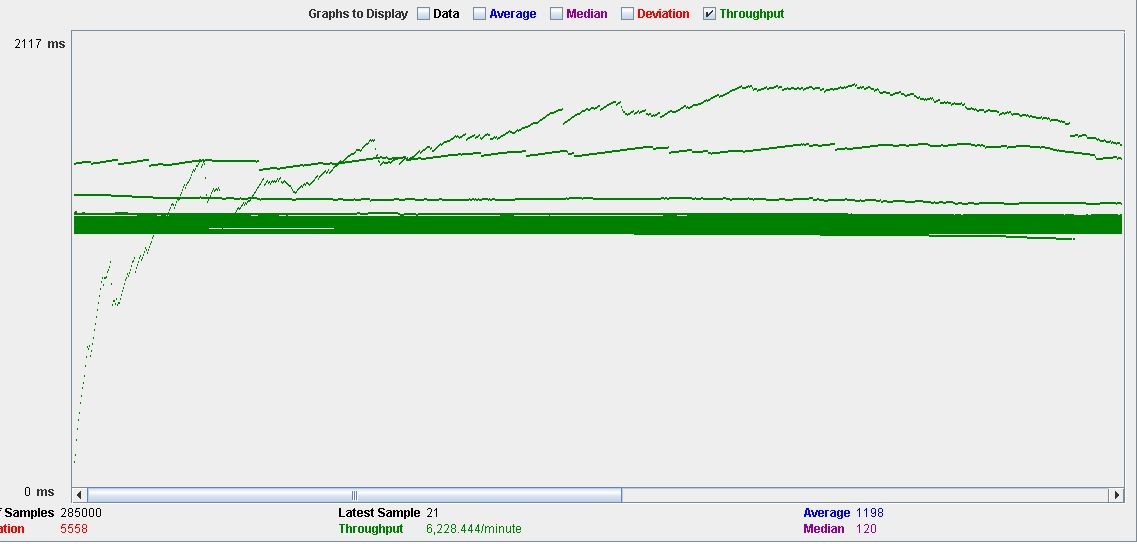
# Browse chorbies

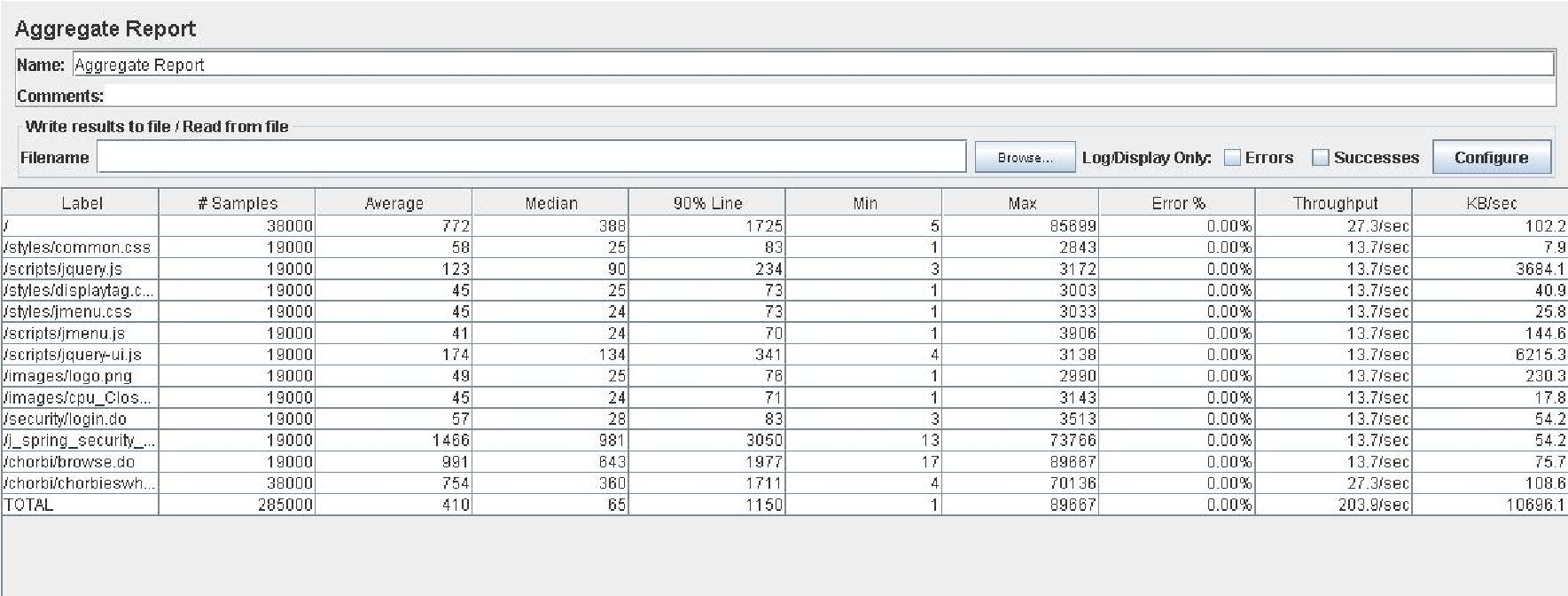


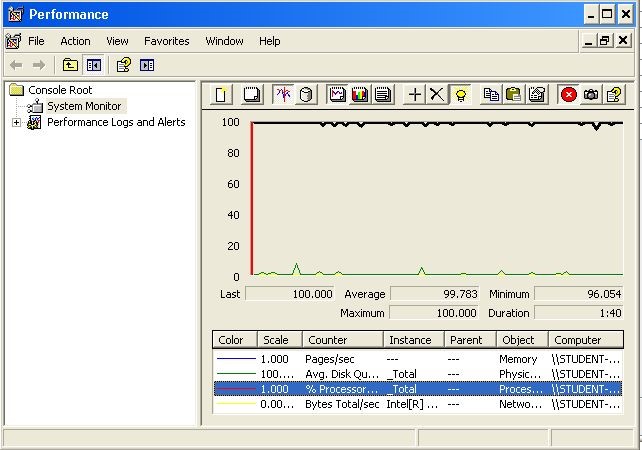




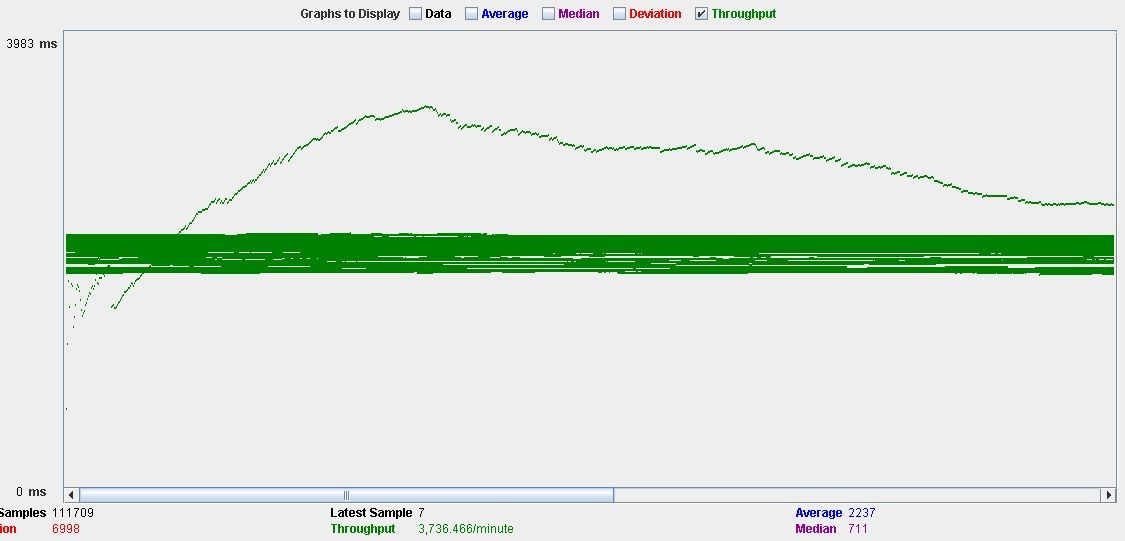
# Browse chorbies who like them

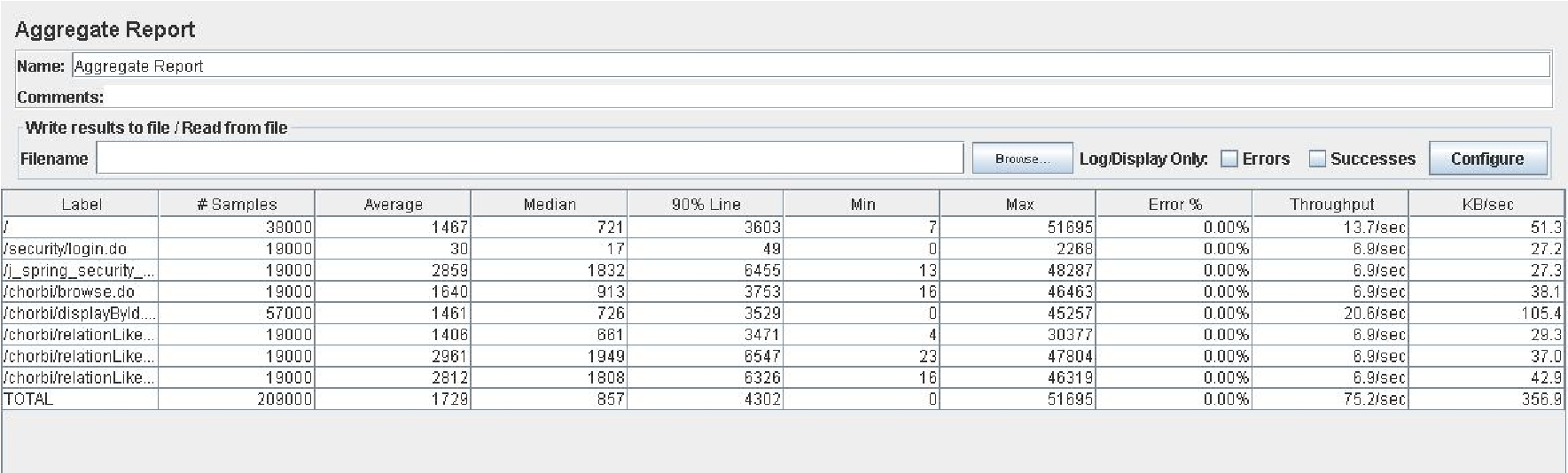


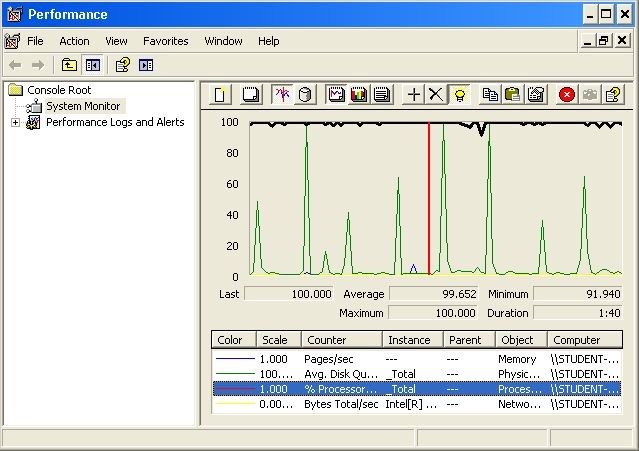




# Like and cancel like

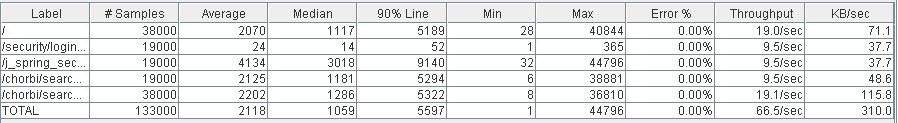


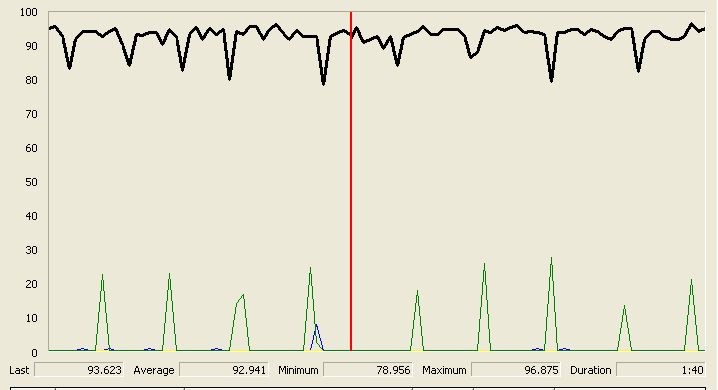




# Edit searchTemplate and browse results

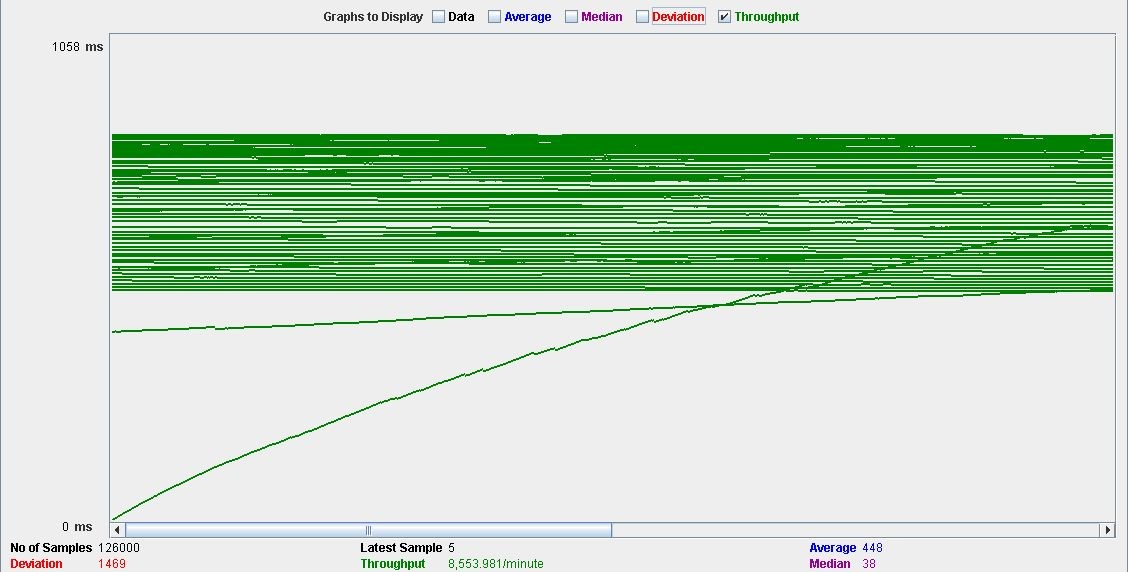


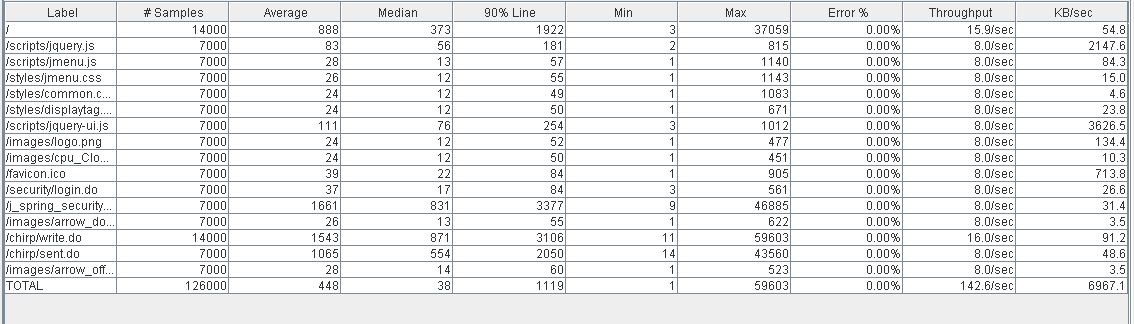


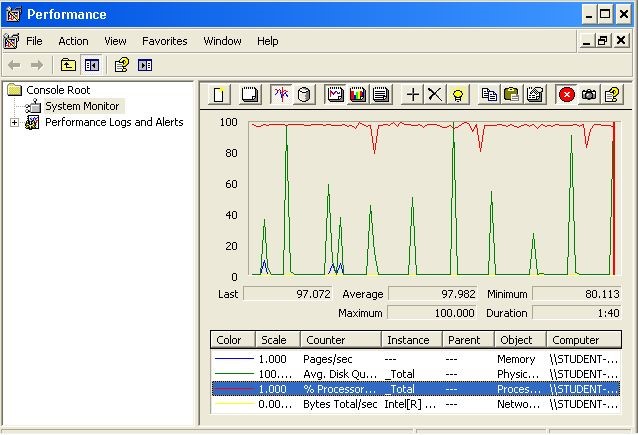


# Write chirp

Para realizar el test de write chirp hemos tenido que realizarlo con 100 usuarios y 70 loops, con 190 funcionaba correctamente pero al no borrar los mensajes que se iban creando los tiempos aumentaban mucho al tener que cargar la lista de mensajes enviados y terminaba quedándose la máquina colgada.



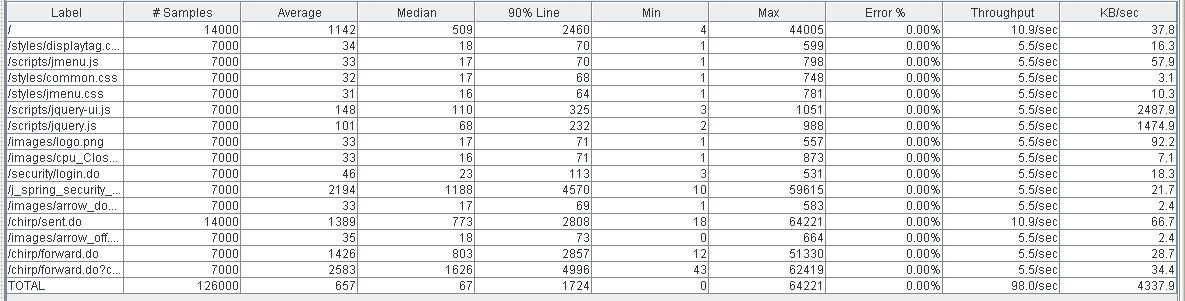


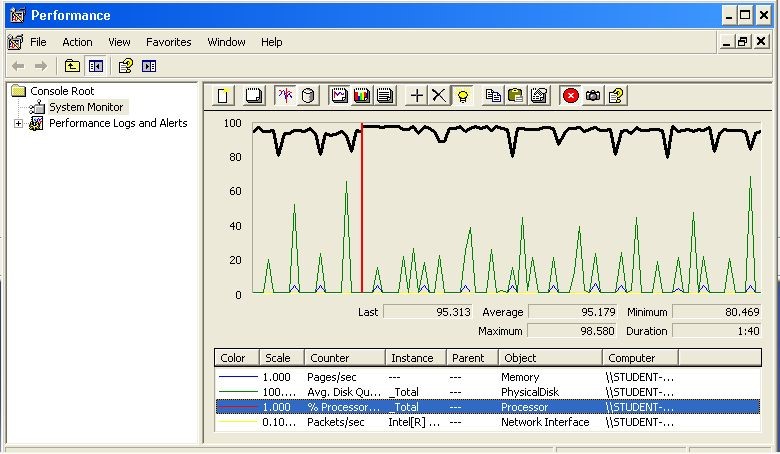


# Forward chirp

Para realizar el test de forward chirp hemos tenido que realizarlo con 100 usuarios y 70 loops, con 190 funcionaba correctamente pero al no borrar los mensajes que se iban creando los tiempos aumentaban mucho y terminaba quedándose la máquina colgada.



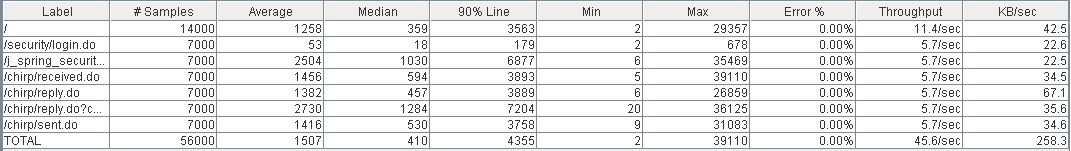


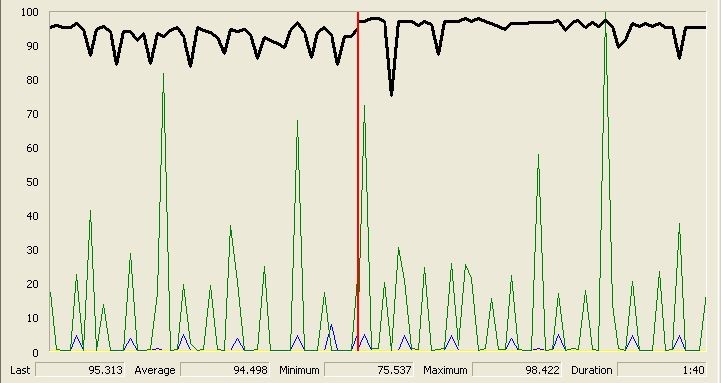


# Reply chirp

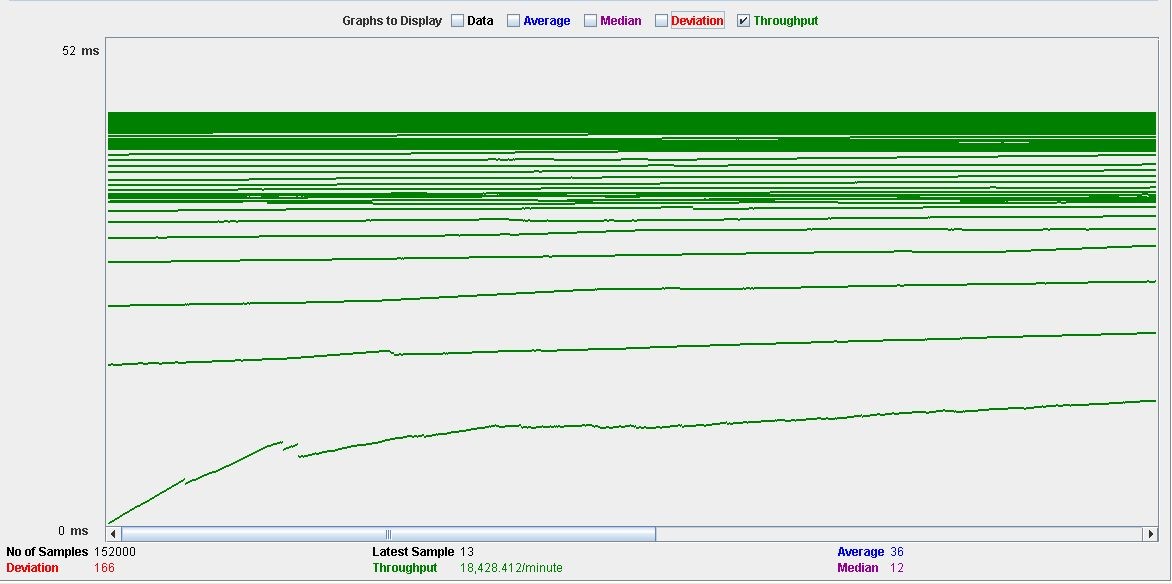
Para realizar el test de reply chirp hemos tenido que realizarlo con 100 usuarios y 70 loops, con 190 funcionaba correctamente pero al no borrar los mensajes que se iban creando los tiempos aumentaban mucho y terminaba quedándose la máquina colgada.

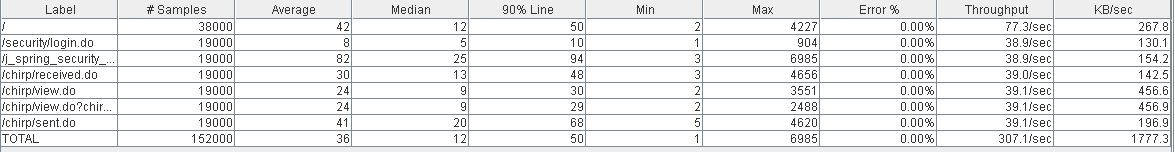


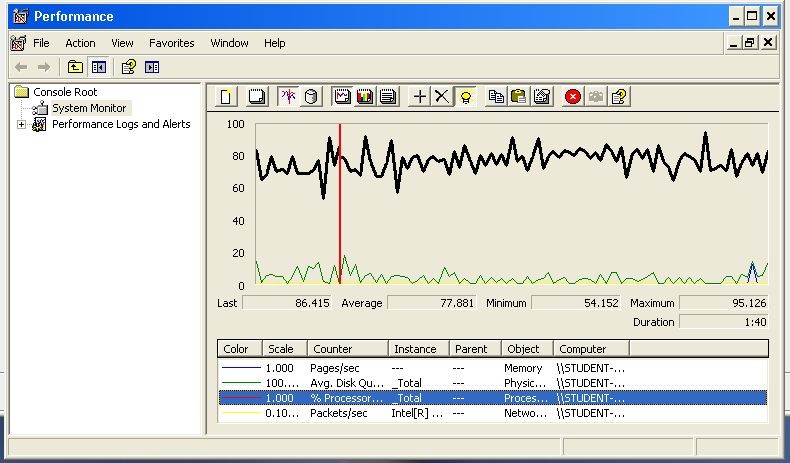




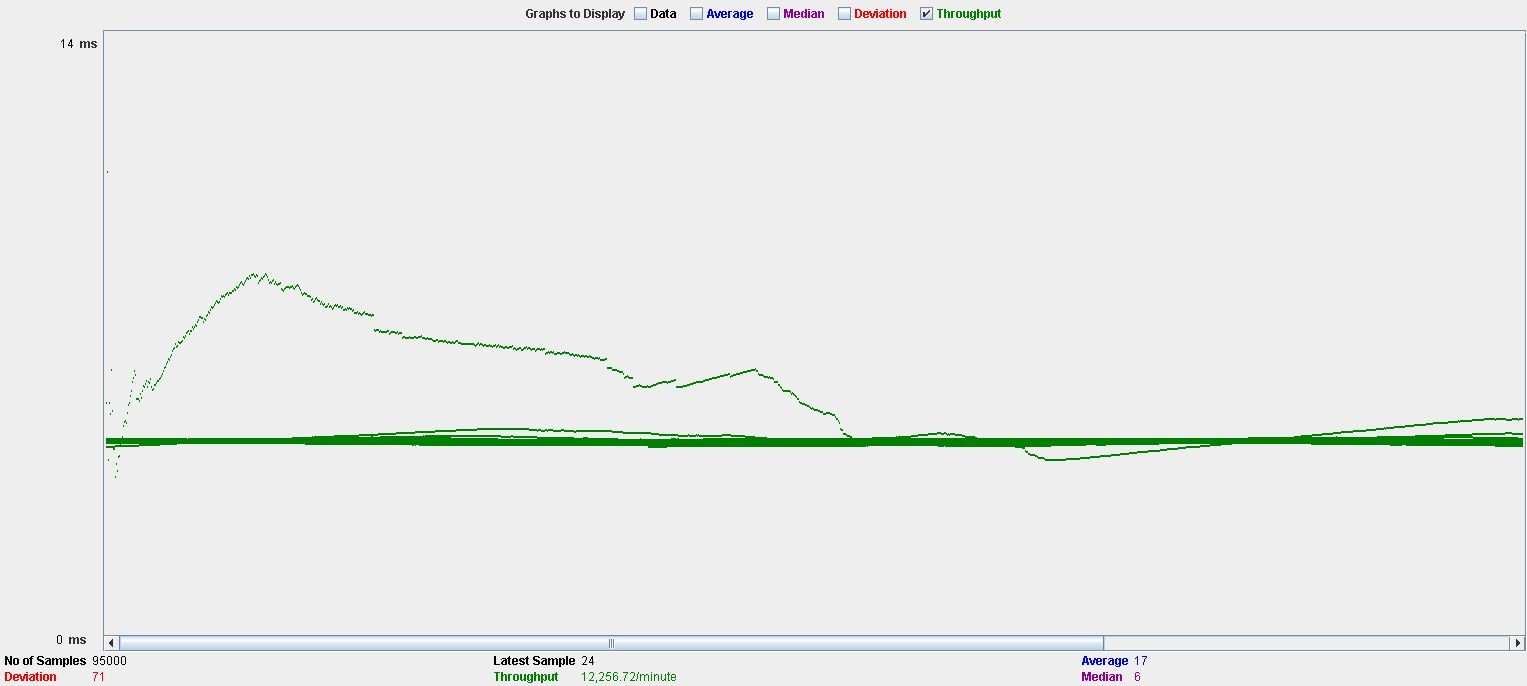
# Delete chirp

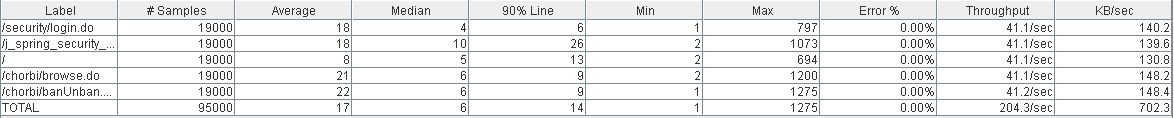


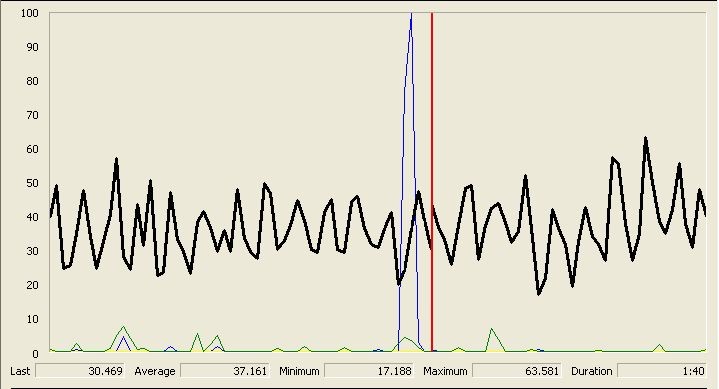




# Ban and unban chorbi

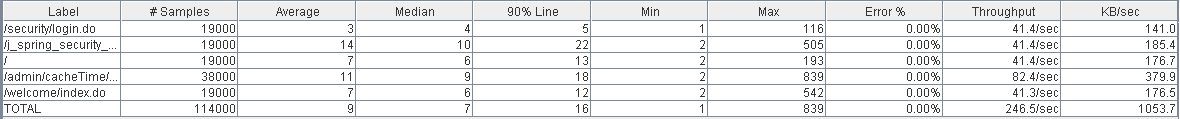


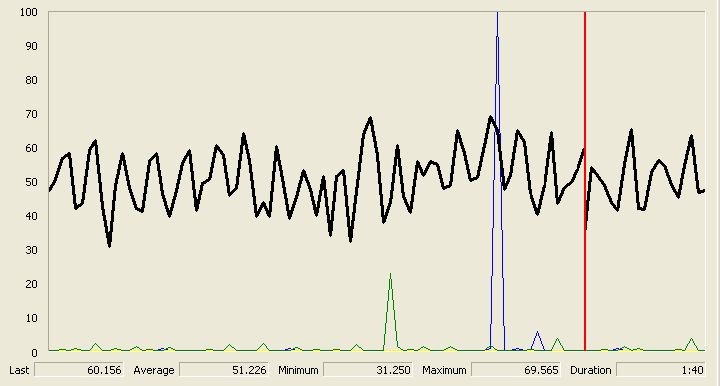




# Edit cache time

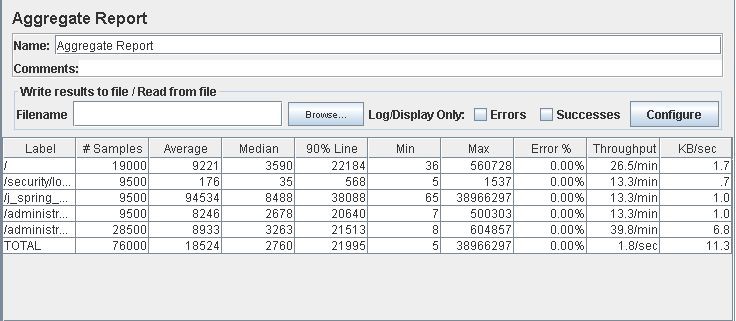


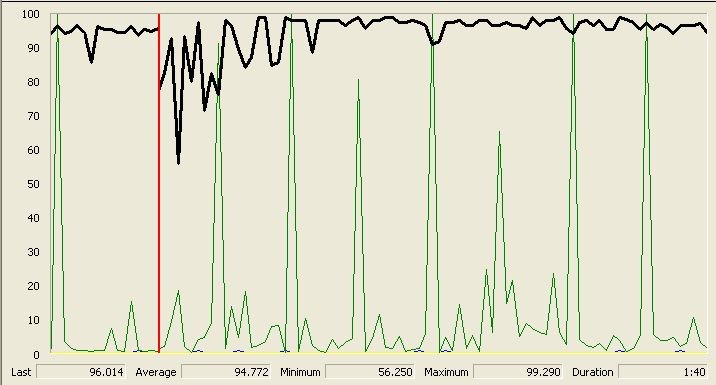




# Change banners

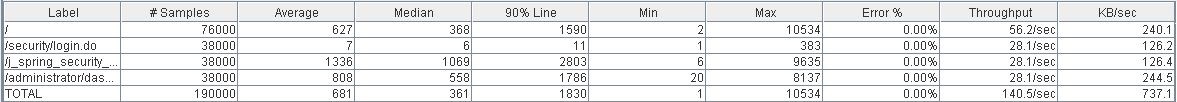






# Dashboards







# Conclusiones

Como podemos observar en todos los test, el procesador es nuestro cuello de botella, si lo reemplazamos por un procesador más potente podríamos aumentar la cantidad de usuarios simultáneos.